Computers and Mathematics with Applications 62 (2011) 3504–3523

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

A path switching scheme for SCTP based on round trip delays

Fang-Yie Leu [a,*], Fenq-Lin Jenq [b], Fuu-Cheng Jiang [a]

[a] *Department of Computer Science, Tunghai University, Taichung, Taiwan*
[b] *Department of Electronic Engineering, Far East University, Tainan, Taiwan*

ARTICLE INFO

Article history:
Received 30 March 2011
Received in revised form 30 July 2011
Accepted 28 August 2011

Keywords:
SCTP
Congestion control
Retransmission
Bandwidth
Path switch
Round trip delay

ABSTRACT

Due to the rapid development of network applications, today the Internet plays an important role in our everyday life. Users hope that the network is always speedy enough to help them access the Internet without any delay. But the real situation is far from the ideal case. In the future, network researchers will continuously improve the network speed, and try to develop networks that are robust, without any crashes or packet loss. In this paper, we propose an aggressive path switching scheme for SCTP. Before data transmission, the scheme selects the fastest path as the primary path to transmit packets. When the path fails or transmission quality is poor, this scheme evaluates alternate paths, and selects the one with the best quality as the new primary path to substitute for the original one. After that, packets are delivered through the new path. Several factors are considered in the evaluation, including bandwidth, encryption/decryption, size of the congestion window, retransmission policy, routing policy, etc.

1. Introduction

In recent years, many mobile and fixed hosts are increasingly equipped with multiple network interfaces [1] which provide two end nodes of a communication connection/association with multiple paths to enhance packet delivery reliability and service availability. This is an important issue particularly for those systems that need very reliable transmission support. Stream control transmission protocol (SCTP), which provides a multi-homing feature, is presently a protocol that meets the requirement of multiple network interfaces. That is why its importance both in wired and wireless communication is greater every day, and its applications have also been widely deployed and quickly developed. Leu [2] employed SCTP as a key mechanism of network mobility to achieve a seamless handover for delivering multimedia data. Noonan et al. [3] proposed a delay sensitive SCTP which evaluates voice traffic between multi-homed hosts and chooses the lowest delay path to demonstrate performance improvements.

To take full advantage of multi-homing, many current studies are addressing the issue of how to select the best network interface and transmission path to efficiently transmit packets. Dahal and Saikia [4] proposed a scheme, called Switch Path on Congestion, to determine whether a handover from the current primary path to an alternate path is necessary or not. Kelly et al. [5] introduced a delay-centric handover by periodically measuring path delays. Ribeiro and Leung [1] raised a minimum delay path selection scheme to select the lowest delay paths for both directions of communication between sender and receiver. Noonan et al. [3] proposed a scheme that offers the benefit of performing the handover based on measured path delays. Other modified SCTP versions can be found in [6,7]. Al-kaisan et al. [8] stated that congestion control algorithms are unable to prevent congestion collapse and unfairness created by applications that are unresponsive to network congestion.

* Corresponding author. Tel.: +886 930 069 809; fax: +886 4 2359 1567.
E-mail addresses: leufy@thu.edu.tw (F.-Y. Leu), fljenq@ms41.hinet.net (F.-L. Jenq), admor@thu.edu.tw (F.-C. Jiang).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.08.066

Generally, the above-mentioned approaches switch paths and improve SCTP performance all based on measured round trip time (RTT) and layer-four features, e.g., adjusting the size of the congestion window. However, our opinion is that SCTP improvement should not be limited to RTT measurements and transport layer functions, implying that the factors the current SCTP systems consider are only a part of SCTP's performance-affecting factors. It also means that the performance can be further improved. In fact, a packet travels through several network layers before it arrives at its destination. We consult the OSI model [9] as the reference model. When a transmission starts, packets flow from the application layer to the physical layer, and then go across switches/routers or mobile routers. When the packets arrive at the receiver, they go in the reverse direction from the physical layer to the application layer. This is a complicated transmission process in which the transport layer plays an important role in handling flow control. However, if other factors, like current network bandwidth and packet drop rates on intermediate nodes/routers between sender and receiver, can be involved [10], its performance will be further improved.

Hence, in this study, we develop a new path switching scheme for SCTP, called the *p*ath *s*election *a*nd *s*witching *p*rocess (PSASP), which when the current primary path fails or its transmission quality is poor chooses the best path for SCTP by evaluating mechanisms and activities that influence SCTP transmission efficiency, including the size of encrypted/decrypted data [6], size of the congestion window [11,12], retransmission policies [13], length of a routing path [14], a packet's RTT [4], network delays [1], hardware speed and bandwidth, etc., aiming to improve the performance of the SCTP protocol. These mechanisms and activities are dispersed in layers of the OSI model. For example, routing is a layer-three task, and hardware speed is a layer-one concern. In this study, we also formally analyze a path's delivery delay by dealing with these probable factors, and propose a path switching scheme based on evaluation results of the related mechanisms and activities. Further, among these factors, a factor may be affected by others. For example, current available bandwidth is affected by the size of the sender's congestion window. In other words, this is a complicated analytical task. Experimental results show that this scheme can truly select the best path. In the following, no matter the concerned facilities are routers or mobile routers, we call them routers to simplify the description.

The contributions of this research are as follows:

(1) The PSASP evaluates cross-layer mechanisms and activities to select a primary path for the SCTP.
(2) We derive PSASP's cost model, including the processing delay, transmission delay, propagation delay and queuing delay, each of which is evaluated based on the cross-layer mechanisms and activities.
(3) We calculate the total cost for the PSASP when k retransmissions have been experienced given a path's retransmission probability, $k = 0, 1, 2, \dots, n$.

This paper is organized as follows. Section 2 introduces relevant background and related work. Section 3 describes our system architecture. The experimental results are presented in Section 4. Section 5 concludes this article and addresses our feature work.

2. Background and related work

2.1. SCTP

The SCTP inherits features and attributes from the TCP, but provides new features for users [15], including multi-homing, multi-streaming, heartbeat, four-way handshake, and chunk bundling.

(1) Multi-homing: with this, the SCTP establishes an association between sender and receiver before transmitting packets. An association often contains multiple paths, each of which is an ip-to-ip connection, i.e., this protocol needs multiple IPs. Initially the SCTP chooses a path as the primary path to transmit packets. When transmission quality is poor, it chooses the secondary path, known as alternate path, to substitute for the primary path. With multi-homing, SCTP transmission is more reliable than that of TCP and UDP.
(2) Multi-streaming: this divides a path into multiple subpaths, called streams. All streams are independent of each other in transmission. Before data transmission, SCTP defines a number of streams and assigns packets to streams for transmission to prevent the head of line problem [16].
(3) Heartbeat: this is implemented for each node to periodically send packets telling other nodes that it is still active. Through heartbeats, a node can know which paths are currently available.
(4) Four-way handshake: this is used to establish a connection. Before data transmission, the sender sends an INIT to the receiver. The receiver on receiving the INIT responds with an INIT-ACK which includes a state cookie and connection information, neither saving state information, nor allocating resources for the connection. Next, the sender replies with a corresponding COOKIE-ECHO to confirm the state cookie. After the confirmation, the receiver replies with a COOKIE-ACK. After that, an association is established and the sender can transmit data to the receiver. Meanwhile, the receiver allocates cpu time and memory capacity to the association.
(5) Chunk bundling: this is related to the SCTP packet format. A SCTP packet includes control chunks and data chunks. Control chunks carry information for SCTP controlling. Data chunks convey data messages. The SCTP can bundle several small chunks into a big one, or vice versa. However, the packet size cannot in any circumstance exceed the maximum transmission limit.

2.2. The SCTP variations and applications

There are several SCTP variations [17,18]. Mobile-SCTP (mSCTP) [17], an extension of SCTP, is used for mobility management in a wireless environment. It allows an endpoint to add, delete and change IPs by sending address configuration (ASCONF) messages to its peer while their SCTP association is still active. Leu [2] used the mSCTP protocol to design a wireless handoff scheme by exploiting the SCTP multi-homing feature.

Satellite networks are global internet that provides broadband transmission, television, and navigation services. Fu et al. [18] investigated and evaluated the SCTP features to increase the satellite network performance. Kim et al. [19] used the SCTP to support the real-time Internet Protocol Television (IPTV) which has been regarded as one of the applications in the next generation networks. Through multi-streaming, the streams can dispatch stream 0 as service manager, stream 1 as channel 1, …, stream n as channel n. All the channels can easily transfer packets through different stream identifiers to reduce the impact of the head-of-line blocking problem.

2.3. Related work and influential factors

According to previous studies [4,6,11–14,20], network transmission is influenced by several factors. Yang et al. [6] mentioned that encryption, due to requiring additional overhead, makes a data chunk include much more information than transmitting plain text does. The overhead consumes extra packet processing time and transmission time. Generally, a relatively higher security level often generates more overhead than a lower level one does. Kim et al. [11] pointed out that different congestion-window increasing/shrinking polices result in different throughputs. Fallon et al. [13] described how retransmission policies, e.g., different parameters such as Path.Max.Retrans (PMR) and Retransmission Time-Out (RTO), cause different failover performance.

Routing policies, e.g., static and dynamic, also affect transmission performance since different policies select different paths for data transmission. Hassan et al. [14] analyzed two routing protocols: proactive (table-driven) and reactive (on-demand). Proactive protocols, such as Destination-sequence Distance-vector Routing (DSDV) [21], maintain routing information by periodically exchanging routing-table contents with neighbors, whereas reactive protocols, such as Ad hoc On Demand Distance Vector (AODV) [22] and Dynamic Source Routing (DSR) [23], build routing paths when they need to route packets.

Dahel and Saikia [4] stated that round-trip time (RTT) which responds to the current available bandwidth of a path is helpful in determining how to adjust the congestion window and perform path switch during data transmission. The proposed RTT based congestion avoidance (RBCA) Scheme, which calculates RTT on receipt of each SACK, uses the Timestamp option, an added chunk, to adjust *cwnd*, and changes its window size by calculating *CwndIncr* where $CwndIncr = DSize * \frac{\mathrm{RTT}_{threshold} - \mathrm{RTT}}{\max(\mathrm{RTT}, \mathrm{RTT}_{threshold})}$, in which $DSize = \min$ (number of newly ACKed bytes in a SACK, maximum packet size), and the $\mathrm{RTT}_{threshold} (= \mathrm{RTT} * F)$ is a pre-calculated parameter where $F > 1$. $\mathrm{RTT}_{threshold}$ is used to judge the length of RTT. The increment may be positive or negative depending on the values of RTT and $\mathrm{RTT}_{threshold}$. The RTT value is monitored for every SACK. However, the authors only dealt with queuing delay. Other delays, e.g., propagation delay and transmission delay, are considered as constants, particularly for the case when the congestion window shrinks. In the proposed Switch Path on Congestion scheme, when $\mathrm{RTT}_{t3} > \mathrm{RTT}_{t2} > \mathrm{RTT}_{t1} > \mathrm{RTT}_{threshold}$, a new primary path will be selected where $t_3 > t_2 > t_1$.

Ribeiro and Leung [1] stated that symmetric paths and asymmetric paths perform differently. The delays of asymmetric paths are usually shorter because they can choose the path with the lowest delay to transmit packets. Al-kaisan et al. [8] presented another version of the SCTP, called the optimized SCTP, which modified congestion control policy to improve SCTP performance. Once a packet has been retransmitted by the fast retransmission procedure, it is marked as ineligible for retransmission until a transmission times out. Then, it ignores the same lost SACK messages for the current time period. In other words, only the first detection of a lost packet will cause the path variables to be changed. Further, the authors proposed a congestion control approach, with which a detection on packet loss will cause slow reduction of *cwnd*, i.e., $cwnd \leftarrow cwnd - [0.05 * cwnd]$ instead of reducing it by half where *cwnd* stands for congestion window. When congestion is not serious, this approach can achieve better performance. However, when congestion is severe, e.g., transmission quality of the primary path is very poor or when it is broken, its packet loss rate will be higher. In this case, choosing an alternate path by exploiting the multi-homing feature of the SCTP can often effectively improve data transmission efficiency [1].

The standard SCTP uses the heartbeat to detect a path's current condition and mark the path as active or inactive. When the primary path is broken, it will be marked as inactive, and the standard SCTP will select the next available active path to be the new primary path for data transmission. The drawback of the algorithm is that the selected path may not be the fastest one. So how to select the fastest path before path switching is an important issue.

In fact, the round-trip time is a good method to evaluate alternate paths. But, analyzing round-trip delay is a complicated task since it consists of many path-performance affecting factors which are dispersed among different network layers. However, it actually reflects the real condition of a path because a packet and its acknowledgement are delivered through the path. In this study, we will analyze how the factors affect path performance. Based on the analysis, we can then select the best path, i.e., the one with the widest current bandwidth, as the primary path to deliver messages. This can decrease the retransmission probability and improve the total transmission performance.

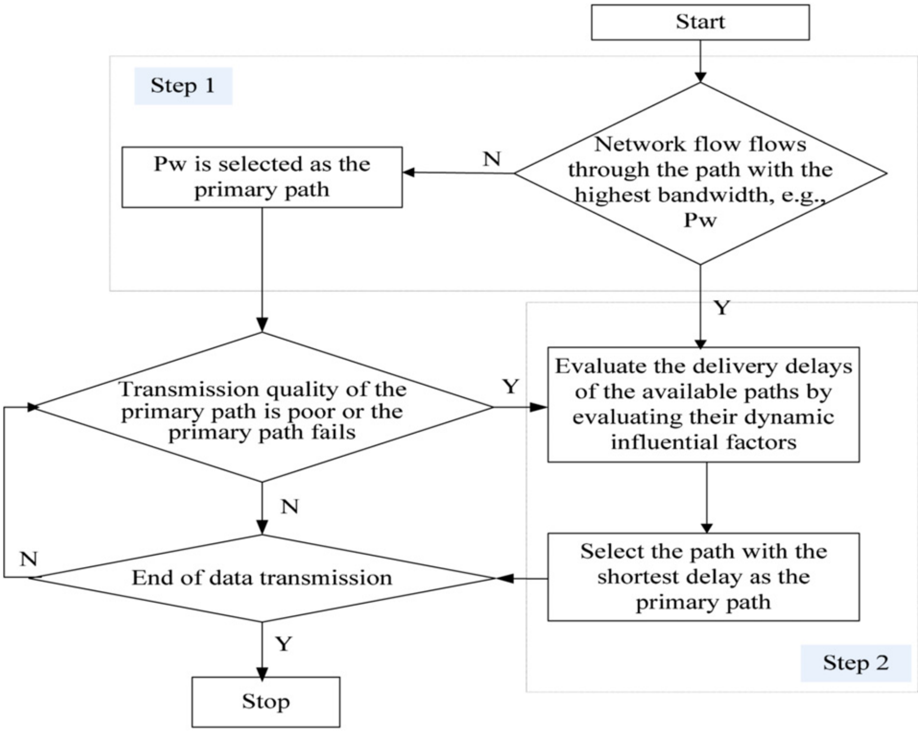

Fig. 1. The flow chart of the PSASP.

3. The proposed scheme

In a multi-homing environment, the PSASP has two main steps in selecting the best path for an SCTP association. Step 1 is selecting an initial primary path. Before data transmission, the PSASP first checks to see whether or not any network flow flows through the path with the widest bandwidth. If not, the path will be selected as the initial primary path. Otherwise, the PSASP enters Step 2 which evaluates performance for all paths of the association and then selects the best one as the primary path. In addition, when transmission quality is poor or the primary path fails, the PSASP will also invoke the Step-2 process. But, this time, only the available paths of the association are evaluated. In this process, dynamic influential factors which will be described later are employed to compute the packet delivery delay of a transmission path. The one with the minimum delay or default path will be selected as the initial or the new primary path. In the following, we assume that (1) the bandwidth of an association and that of each path involved are known; (2) initial bandwidth = current_available bandwidth + occupied_bandwidth; (3) the packet arrival rate of each path segment along a path, e.g., the path segment between nodes i and $i + 1$, follows a Poisson distribution. The flow chart of the PSASP is shown in Fig. 1.

3.1. Dynamic factors

The following mechanisms and activities, including switchover/retransmission policies, size of encrypted/decrypted data, size of the congestion window and round-trip delay, are considered as key factors in selecting a primary path.

3.1.1. Switchover/retransmission policies

There are two main factors that strongly influence retransmission policies. One is PMR which is the maximum retransmission count of a path. The other is RTO which is the counted time of a retransmission period. Fallon et al. [13] claimed that PMR and RTO should both be considered before an appropriate switchover/retransmission policy can be ensured. That is, when the retransmission timer exceeds the RTO, an underlying packet will be retransmitted. When the retransmission count is over the PMR (i.e., a path's transmission-failure count $\geq$ PMR + 1), implying the quality of the path is poor or the path fails, a new path will be selected, and the SCTP will switch over to the new path to continue delivering packets for the sender. Often, the recommended value of PMR is 4 [5] or 5 [13], and those of RTO.initial, RTO.min and RTO.max are, respectively, 3 s, 1 s and 60 s [24].

3.1.2. Size of encrypted/decrypted data

According to [6], an encrypted packet has a longer delivery delay than its original packet has because of additional processing efforts, such as packet encryption and decryption, and additional transmission overheads. Generally, encrypted packets can be classified into four security levels [6]. Level 0 does not provide any security facilities. Level 1 provides

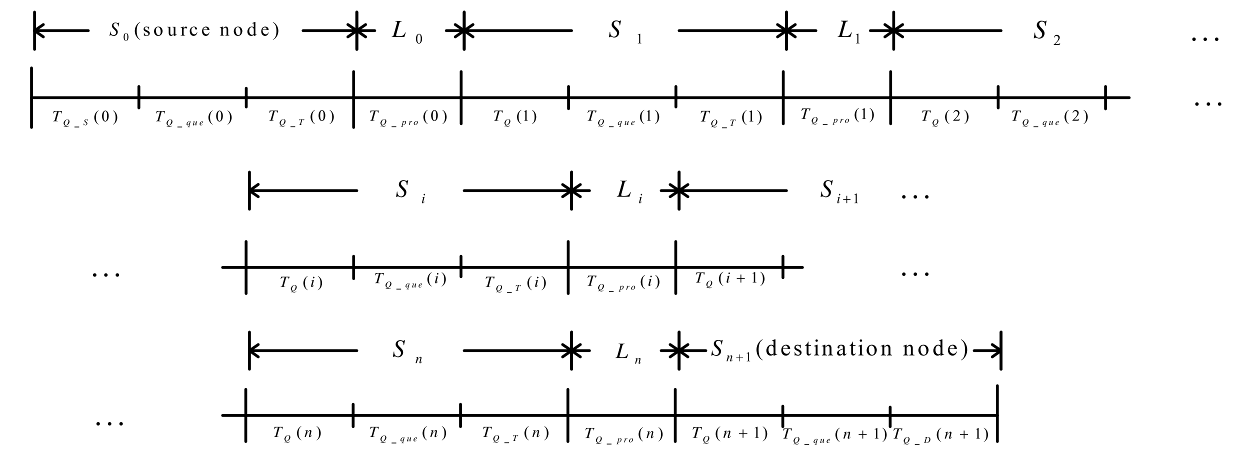

Fig. 2. The timings of a path with $n+2$ nodes ($S_0, S_1, \ldots, S_{n+1}$) in which S_0 is the source node which generates a packet Q. S_0 is also the destination of the corresponding *ACK*. S_{n+1} is the destination node of Q and S_{n+1} is also the source node of the *ACK*.

authentication and integrity checking for established associations. With level 2, only a part of chunks, instead of the whole, is encrypted. Level 3 provides encryption, authentication and integrity checking for all chunks. Higher security levels often have more overheads.

3.1.3. Size of congestion window

According to [11], when the size of a congestion window is relatively larger, implying the transmission path has better quality, and current available bandwidth defined as (initial_bandwidth − traffic_occupied bandwidth) is wider, then a sender can transmit more data per second to a receiver. When the window size is small, it often means the available bandwidth of the path is limited, and the network quality is not good. Once packets are lost, the window size will be reduced to mitigate data flow and shorten packet waiting time. In this case, the SCTP can only use a portion of currently available bandwidth to transmit packets. In this study, we further assume that available bandwidth = (initial_bandwidth − traffic_occupied bandwidth − SCTP_occupied bandwidth) where SCTP_occupied bandwidth is caused by shrunken congestion window size. If packets can be successfully and continuously delivered to the destination, the window size will be slowly enlarged, which is known as a slow start.

3.2. Round trip delay

According to [4], the RTT more accurately reflects real network speeds. Many systems employ it as an important performance parameter. A shorter round trip time implies the network transmission speed is high. Ribeiro and Leung [1] used the round trip time to judge the paths. But, the authors did not analyze details of the delay. In this study, we consider round trip delay as the key performance-measure parameter. The delay can be further divided into transmission, propagation, processing and queuing delays.

3.2.1. The timings of delivering a data packet

In the following, we assume the SCTP association has H paths, and a chosen path contains $n+2$ nodes, including the source node, denoted by S_0, the destination node, denoted by S_{n+1}, and n intermediate nodes (i.e., n routers), denoted by $S_1, S_2, \ldots, S_n$.

In Fig. 2, S_0 first generates a packet (i.e., Q) which will be delivered to S_1 through link 0, denoted by L_0. The time required to generate and encrypt a packet by S_0 is $T_{Q\_S}(0)$. If S_0's packet generating speed is higher than the delivery speed, packets will be queued in S_0's message buffer. The time a packet waits in S_i's message buffer is $T_{Q\_que}(i)$. All $n+2$ nodes have their own queues. That is why the indexes of $T_{Q\_que}(i)$ are from 0 to $n+1$. The time required to transmit Q by S_i is $T_{Q\_T}(i)$. Only $S_0, S_1, \ldots, S_n$ transmit packets to their immediate downstream nodes. Therefore, the indexes of $T_{Q\_T}(i)$ are between 0 and n. Once Q is delivered by S_i, it will travel through L_i to S_{i+1}. The time required by a bit to propagate from S_i to S_{i+1} through L_i is $T_{Q\_pro}(i)$. Q should travel through $n+1$ links ($L_0, L_1, \ldots, L_n$) before it can arrive at S_{n+1}. So, the indexes of $T_{Q\_pro}(i)$ are between 0 and n. When the first bit of Q arrives at S_i, S_i starts receiving Q. The time required to receive Q at S_i is $T_Q(i)$. Only $S_0, S_1, \ldots, S_{n+1}$ receive Q from their upstream nodes. So, the indexes of $T_Q(i)$ are between 1 and $n+1$. Lastly, $T_{Q\_D}(n+1)$ is S_{n+1}'s decryption time. The above mentioned variables are listed and described in Table 1.

3.2.2. The timings for delivering an ACK

The timings for delivering an *ACK* and their indexes are shown in Fig. 3. The definitions of these terms are listed in Table 2. Their descriptions are similar to those of delivering Q with Q being substituted by the *ACK*.

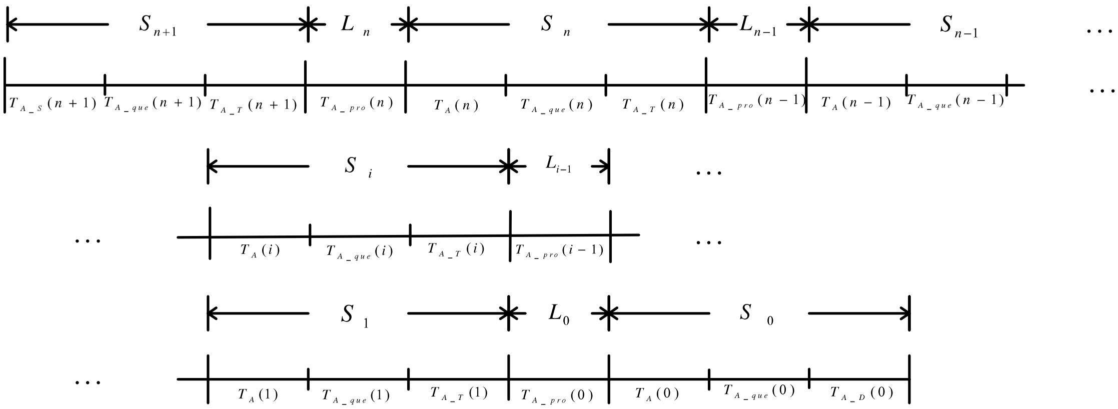

Fig. 3. The timings in delivering an *ACK*.

Table 1
Definitions of terms involved in delivery of data packet Q.

Term	Description
$T_{Q\_S}(0)$	The time required by source node S_0 to generate and encrypt a packet Q, i.e., processing delay at S_0
$T_{Q\_que}(i)\ i = 0, 1, 2, \ldots, n+1$	The time Q waits in S_i's message queue, i.e., waiting time
$T_{Q\_T}(i)\ i = 0, 1, 2, \ldots, n$	The time required by S_i to transmit Q from its first bit to last bit, i.e., transmission delay
$T_{Q\_pro}(i)\ i = 0, 1, 2, \ldots, n$	The time required by Q to propagate from S_i to S_{i+1} through link L_i. It is defined as the time from when a bit is sent out by S_i to the time point when the bit arrives at S_{i+1}, i.e., propagation delay
$T_Q(i)\ i = 1, 2, \ldots, n+1$	The time required by S_i to receive Q. It is defined as the time period from when Q's first bit arrives at S_i to the time point when Q's last bit arrives at S_i, i.e., receiving delay
$T_{Q\_D}(n+1)$	The time required by destination node S_{n+1} to decrypt Q, i.e., processing delay at S_{n+1}

Table 2
Definitions of terms involved in delivery of an *ACK*.

Term	Description
$T_{A\_S}(n+1)$	The time required by S_{n+1} to generate an *ACK*
$T_{A\_que}(i)\ i = 0, 1, 2, \ldots, n+1$	The time the *ACK* waits in S_i's message queue
$T_{A\_T}(i)\ i = 1, 2, \ldots, n+1$	The time required by S_i to send out the *ACK* from the first bit to the last bit (S_0 is the destination)
$T_{A\_pro}(i)\ i = 0, 1, 2, \ldots, n+1$	The time required by the *ACK* to travel from S_i to S_{i-1}. It is defined as the time from when a bit is sent out by S_i to the time point when the bit arrives at S_{i-1}
$T_A(i)\ i = 1, 2, \ldots, n+1$	The time required by S_i to receive the *ACK*. It is defined as the time period from when the *ACK's* first bit arrives at S_i to the time point when the *ACK's* last bit arrives at S_i
$T_{A\_D}(0)$	The time required by S_0 to process the *ACK*, since S_0 does not decrypt the *ACK*, $T_{A\_D}(0) = 0$

Table 3
The speeds involved in packet processing delays.

Term	Description
Data generating speed	The speed with which a node generates a bit. After the generation of a packet, the packet is ready to be transmitted or encrypted
Encryption speed	The speed with which a node encrypts a bit
Decryption speed	The speed with which a node decrypts a bit
Receiving speed	The speed with which a node receives a bit

3.2.3. Processing delay

Processing delay is the time required to prepare and receive a packet, and encrypt and decrypt SCTP chunks. The purpose of these activities is basically getting the data ready for the next activity, e.g., to be transmitted. Performance of the activities is mainly influenced by hardware processing speed and time complexities of the encryption and decryption algorithms. The items involved include size of encrypted/decrypted data, a node's data generating speed, encryption speed, decryption speed, receiving speed and processing speed. The latter five (i.e., speeds) are described in Table 3. The time required to generate a data packet varies dramatically. For example, a control system on receiving a user command may consume a very long time to perform a complicated computation. A sensor of a wireless sensor network may on the contrary, when detecting environmental changes, spend only a few microseconds to transform the changes to formatted data.

Basically, the data generating speed of a computer system strongly depend on its cpu performance. Ohlendorf et al. [25] presented this expression for cpu processing speed: $\frac{cpu\_count \times cpu\_clock}{clock\_cycles\_per\_instruction} MIPS$ where *cpu_count* is number of cpus that a node has, *cpu_clock* is a cpu's clock rate and *clock_cycles_per_instruction* represents the number of clocks required to finish the execution of an instruction. As an example, Kim and Rixner [26] pointed out the fact that when a TCP connection is established and a packet of the maximum size (1460 bytes) is sent with 100 Mb/s (or 900 Mb/s), the required TCP layer's instruction count is 1286 (1356), which is also the number of instructions required to generate a TCP packet. We can infer that the packet generating speed at S_0, denoted by $gen\_speed(0)$, is

$$gen\_speed(0) = \frac{instructions\_executed\_per\_second}{instructions\_per\_packet} \tag{1}$$

where *instructions_executed_per_second* is the number of instructions executed by the cpu and *instructions_per_packet* is the number of instructions required by the cpu to generate a packet.

To express packet encryption/decryption speed, we use the Advanced Encryption Standard (AES) [27], a symmetric encryption mechanism, as an example, and assume that the encryption speed is equal to decryption speed. The time required to encrypt a bit can be derived from the penalty of the AES data encryption [28] through the Regression Analysis [29]. When the lengths of encryption keys are 128, 192, and 256 bits, we can respectively obtain three linear equations, $y = 18.929x + 500$, $y = 22.5x + 214.29$ and $y = 26.25x + 535.71$, from the experiments we have done before. These equations can be generally expressed by $y = \alpha x + \beta$ where x is length of the encrypted data in kilobytes, y in microseconds is the time required to encrypt the data, and α and β are constants once the encryption key length is given.

(1) *Cost for processing a data packet*

To send Q to S_{n+1}, only S_0 generates and encrypts Q. Hence, we can derive the equation for $T_{Q\_S}(0)$,

$$T_{Q\_S}(0) = \frac{1}{gen\_speed(0)} + \left(\alpha \cdot \frac{size(encrypted\ data\ x)}{10^3} + \beta\right) \cdot 10^{-6} \cdot z \tag{2}$$

where x is the portion of Q that is encrypted, $\left(\alpha \cdot \frac{size(encrypted\ data\ x)}{10^3} + \beta\right) \cdot 10^{-6} \cdot z$ is the encryption cost of AES and z is a decision variable. Let Q' be the encrypted Q. Then, $|Q'| = |Q| +$ encryption overhead. If S_0 does not encrypt Q, then $z = 0$, i.e., the encryption cost $= 0$ and $Q' = Q$. Otherwise, $z = 1$.

A node's receiving speed basically depends on its network interface's current input data rate (receiving rate) and input drop rate. Generally, popular network interface data rates are 10 Mbps, 100 Mbps and 1 Gbps. Since $S_1, S_2, \ldots, S_n$ do not encrypt and decrypt Q', based on the definition of $T_Q(i)$, we can derive

$$T_Q(i) = \frac{size(Q')}{rec\_speed(i) - drop\_rate_{in}(i)}, \quad i = 1, 2, \ldots, n+1 \tag{3}$$

where $rec\_speed(i)$ is S_i's receiving speed, and $drop\_rate_{in}(i)$ is S_i's arriving data drop rate, rather than its packet drop rate. Let

$$R_i = rec\_speed(i) - drop\_rate_{in}(i) \tag{4}$$

which is the actual receiving speed of S_i. Let T_{Q_in} be accumulated processing time consumed by the n intermediate nodes to deliver Q'.

$$T_{Q\_in} = \sum_{j=1}^{n} T_Q(j). \tag{5}$$

For the destination node S_{n+1}, the time required to decrypt Q (i.e., $T_{Q\_D}(n+1)$) is

$$T_{Q\_D}(n+1) = \left(\alpha \cdot \frac{size(x)}{10^3} + \beta\right) \cdot 10^{-6} \cdot z. \tag{6}$$

Let

$$T'_{Q\_D}(n+1) = T_Q(n+1) + T_{Q\_D}(n+1) = \frac{size(Q')}{R_{n+1}} + \left(\alpha \cdot \frac{size(x)}{10^3} + \beta\right) \cdot 10^{-6} \cdot z. \tag{7}$$

Let $T_{Q\_processing}$ be the total time for processing Q' in the $n+2$ nodes, $S_0, S_1, \ldots, S_{n+1}$,

$$T_{Q\_processing} = T_{Q\_S}(0) + T_{Q\_in} + T'_{Q\_D}(n+1). \tag{8}$$

(2) *Cost for processing an ACK packet*

Note that an *ACK* is often not encrypted. Based on the definition stated above, we can derive the equation for S_i's processing cost which only includes *ACK's* generation cost.

$$T_{A\_S}(n+1) = \frac{1}{gen\_speed(n+1)} \cdot F \tag{9}$$

where $F < 1$ since an *ACK* is often shorter than Q. So, the required time is also shorter.

$T_A(i)$ can be also derived as

$$T_A(i) = \frac{size(ACK)}{R_i}. \tag{10}$$

The accumulated processing time consumed by the n intermediate nodes is

$$T_{A\_in} = \sum_{j=1}^{n} T_A(j). \tag{11}$$

In S_0, the cost for processing the ACK is decrypting the ACK, so $T_{A\_D}(0) = 0$ since an ACK is not encrypted. Let $T'_{A\_D}(0)$ be the time required by S_0 to receive and process the ACK,

$$T'_{A\_D}(0) = T_A(0) + T_{A\_D}(0) = \frac{size(ACK)}{R_0}. \tag{12}$$

Let $T_{A\_processing}$ be total cost for processing an ACK,

$$T_{A\_processing} = T_{A\_S}(n+1) + T_{A\_in} + T'_{A\_D}(0). \tag{13}$$

Let

$$T_{processing} = T_{Q\_processing} + T_{A\_processing}. \tag{14}$$

3.2.4. Transmission delay

Transmission delay is the time period from when the first bit of Q' is sent out to the time point when the last bit of Q' is transmitted. The items involved in transmission delay include the size of Q' and actual delivery speed, instead of data rate.

S_i's transmission delay,

$$T_{Q\_T}(i) = \frac{size(Q')}{M_i}, \quad i = 0, 1, 2, \ldots, n \tag{15}$$

where M_i is S_i's actual delivery speed which is defined as

$$M_i = data\_rate(i) - drop\_rate_{out}(i). \tag{16}$$

Here, $drop\_rate_{out}(i)$ is the drop rate of S_i's departing data, instead of departing packets.

Let T'_{Q_T} be the transmission delay of the data packet Q',

$$T'_{Q\_T} = \sum_{i=0}^{n} T_{Q\_T}(i) = \sum_{i=0}^{n} \frac{size(Q')}{M_i}. \tag{17}$$

Let T_{Q_T} be the transmission delay caused by the n intermediate nodes while delivering Q'

$$T_{Q\_T} = \sum_{i=1}^{n} T_{Q\_T}(i) = \sum_{i=1}^{n} \frac{size(Q')}{M_i}. \tag{18}$$

Let T'_{A_T} be the transmission delay for delivery of the corresponding ACK.

$$T'_{A\_T} = \sum_{i=1}^{n+1} \frac{size(ACK)}{M_i}. \tag{19}$$

Let T_{A_T} be the transmission delay caused by the n intermediate nodes while delivering the ACK.

$$T_{A\_T} = \sum_{i=1}^{n} \frac{size(ACK)}{M_i}. \tag{20}$$

3.2.5. Propagation delay

Propagation delay of a link L_i connecting S_i and S_{i+1} $(i = 0, 1, 2, \ldots, n)$ is the time period from the time point when a bit of Q' is sent out by S_i to the time point when the bit arrives at S_{i+1} (i.e., the time required by the bit to travel from S_i to S_{i+1}). The items included are the initial bandwidth, occupied bandwidth and S_i's output drop rate $drop\_rate_{out}(i)$.

$$T_{Q\_pro}(i) = \frac{1}{bandwidth(i) - bandwidth_{occupied}(i) - drop\_rate_{out}(i)} = \frac{1}{M_i} \tag{21}$$

where $bandwidth(i)$ and $bandwidth_{occupied}(i)$ respectively represent initial bandwidth and occupied bandwidth of the link $L_i,\ i = 0, 1, 2, \ldots, n$ and

$$data\_rate(i) = bandwidth(i) - bandwidth_{occupied}(i). \tag{22}$$

Based on the definition above, when a packet is transmitted, the total propagation delay T'_{Q_pro},

$$T'_{Q\_pro} = \sum_{i=0}^{n} T_{Q\_pro}(i) = \sum_{i=0}^{n} \frac{1}{M_i}. \tag{23}$$

Let T_{Q_pro} be the propagation delay caused by the n intermediate nodes while delivering Q'.

$$T_{Q\_pro} = \sum_{i=1}^{n} T_{Q\_pro}(i) = \sum_{i=1}^{n} \frac{1}{M_i}. \tag{24}$$

For an *ACK* packet, we assume the propagation delay $T_{A\_pro} = T_{Q\_pro}$ (i.e., input bandwidth and output bandwidth are the same. This is reasonable in a real situation since the network interface is the same one, and the required time has nothing to do with packet length) to simplify the scope of the following analyses.

3.2.6. Queuing delay

The items composing queuing delay include packet arrival and departure rates. Here, we assume the processing mechanism pertaining to a queue is the M/M/1 queuing model [30]. One may conclude that the SCTP's multi-streaming mechanism can be viewed as a multi-server system. This is true. But, from the physical layer viewpoint, the interface is the only mechanism (i.e., server of the queue) that sends out packets, no matter which streams the packets belong to. The arrival rate λ is a function of several independent variables, including the packet size, bandwidth, occupied bandwidth, packet drop rate, length of queue, etc. The departure rate μ (i.e., service rate) is also a function of several independent variables, including hardware processing speed, data rates, etc. Due to involving too many influential factors, it is hard to derive mathematical models for them. But the arrival (departure) rate of a link can be observed on the receiver (sender) side of a path segment.

Based on queuing theory, $T_{Q\_que}(i) = \frac{\lambda_i}{\mu_i(\mu_i - \lambda_i)}$ [30]. Let T'_{Q_que} be the total queuing delay of the current path, $T'_{Q\_que} = \sum_{i=0}^{n+1} \left(\frac{\lambda_i}{\mu_i(\mu_i-\lambda_i)}\right)$, which does not contain service time (i.e., transmission time) and is derived under the assumption that no packets are dropped upon arriving and departing. If we consider actual arrival and departure rates, and assume that they follow a Poisson distribution, then

$$T_{Q\_que}(i) = \frac{R'_i}{M'_i(M'_i - R'_i)} \tag{25}$$

$$T'_{Q\_que} = \sum_{i=0}^{n+1} T_{Q\_que}(i) = \sum_{i=0}^{n+1} \left(\frac{R'_i}{M'_i(M'_i - R'_i)}\right) \tag{26}$$

where R'_i and M'_i are respectively S_i's actual packet arrival and departure rates, rather than data arrival and departure rates, and

$$R'_i = \frac{R_i}{size(Q')} \tag{27}$$

$$M'_i = \frac{M_i}{size(Q')}. \tag{28}$$

Let T_{Q_que} be the queuing delay generated by the n intermediate nodes which process Q'

$$T_{Q\_que} = \sum_{i=1}^{n} T_{Q\_que}(i) = \sum_{i=1}^{n} \left(\frac{R'_i}{M'_i(M'_i - R'_i)}\right). \tag{29}$$

For an *ACK* packet, $T_{A\_que} = T_{Q\_que}/F$ where $F = size(Q')/size(ACK)$, but we assume $T_{A\_que} = T_{Q\_que}$ to simplify the scope of the following analyses.

3.3. Total cost without retransmission

When the initial primary path fails, there are two methods to choose a new primary path. One is to evaluate the remaining H-1 paths one by one, and sort the paths based on their evaluation results. The one with the highest performance is then selected as the new primary path. The other method is comparing two arbitrary paths, e.g., paths q and r. The one with higher performance, e.g., q, will be chosen. We then select an uncompared path as the new r, and compare q and r again. This procedure repeats until all paths are compared. Then, the one with the highest performance is selected. With either method, H-1 paths will be evaluated. But, using the second approach, we can omit the evaluation of many items, e.g., S_0's and S_{n+1}'s costs, since the source nodes and destination nodes of two paths, e.g., paths q and r, belonging to the same association are themselves the same, and the two paths deliver the same packet Q/Q' and ACK. The cost difference CD_{qr} between the two paths only results from involving different numbers of intermediate nodes and different intermediate nodes (note that some of the nodes may be the same).

Let TC be the total cost of packet delivery without retransmission,

$$TC = T_{processing} + (T_{Q\_T} + T_{A\_T}) + (T_{Q\_pro} + T_{A\_pro}) + (T_{Q\_que} + T_{A\_que}). \tag{30}$$

According to Eqs. (5), (11), (18), (20), (24) and (29), the cost difference between TC_q and TC_r is

$$\begin{aligned} CD_{qr} &= TC_q - TC_r = (T^q_{Q\_in} - T^r_{Q\_in} + T^q_{A\_in} - T^r_{A\_in}) \\ &\quad + (T^q_{Q\_T} - T^r_{Q\_T} + T^q_{A\_T} - T^r_{A\_T}) + 2(T^q_{Q\_pro} - T^r_{Q\_pro}) + 2(T^q_{Q\_que} - T^r_{Q\_que}). \end{aligned} \tag{31}$$

From Eqs. (3)–(5), (10) and (11), we can see that the expression $(T^q_{Q\_in} - T^r_{Q\_in} + T^q_{A\_in} - T^r_{A\_in})$ is a function of R_i, $i = 1, 2, \ldots, n_q + n_r$, once $size(Q')$ is given where n_q and n_r are respectively the numbers of path q's and path r's immediate nodes. Similarly, based on Eqs. (18) and (20), the expression $(T^q_{Q\_T} - T^r_{Q\_T} + T^q_{A\_T} - T^r_{A\_T})$ is a function of M_i, $i = 0, 1, 2, \ldots, n_q + n_r$. Based on Eqs. (24) and (29), the remaining two expressions are respectively functions of M_i, R'_i and M'_i. Let $CD'_{qr} = |CD_{qr}|$, and assume that $TC_q > TC_r$, then

$$\begin{aligned} CD'_{qr} &= \sum_{j=1}^{n_q} \frac{size(Q')}{R_{j,q}} - \sum_{k=1}^{n_r} \frac{size(Q')}{R_{k,r}} + \sum_{j=1}^{n_q} \frac{size(ACK)}{R_{j,q}} - \sum_{k=1}^{n_r} \frac{size(ACK)}{R_{k,r}} \\ &\quad + \sum_{j=1}^{n_q} \frac{size(Q')}{M_{j,q}} - \sum_{k=1}^{n_r} \frac{size(Q')}{M_{k,r}} + \sum_{j=1}^{n_q} \frac{size(ACK)}{M_{j,q}} - \sum_{k=1}^{n_r} \frac{size(ACK)}{M_{k,r}} \\ &\quad + 2\left(\sum_{j=1}^{n_q-1} \frac{1}{M_{j,q}} - \sum_{k=1}^{n_r-1} \frac{1}{M_{k,r}}\right) + 2\left(\sum_{j=1}^{n_q} \frac{R'_{j,q}}{M'_{j,q}(M'_{j,q} - R'_{j,q})} - \sum_{k=1}^{n_r} \frac{R'_{k,r}}{M'_{k,r}(M'_{k,r} - R'_{k,r})}\right) \end{aligned} \tag{32}$$

where $R_{j,q}$ ($R_{k,r}$) is actual receiving speed of $S_{j,q}$ (i.e., node j on path q) (of $S_{k,r}$ (i.e., node k on path r)), $M_{j,q}$ ($M_{k,r}$) is actual delivery speed of $S_{j,q}$ (of $S_{k,r}$), and $R'_{j,q}$ and $M'_{j,q}$ ($R'_{k,r}$ and $M'_{k,r}$) are respectively the actual packet arrival rate and departure rate of $S_{j,q}$ (of $S_{k,r}$).

Since an ACK is a packet of fixed length, given an encrypted packet Q' and an association that has two paths, q and r of lengths $n_q + 2$ and $n_r + 2$, respectively, from Eqs. (4), (16), (27) and (28), $R_{j,q}, R_{k,r}, M_{j,q}, M_{k,r}$ are unknown. In turn, from Eqs. (4) and (6), we can see that only ($rec\_speed(j, q)$, $drop\_rate_{in}(j, q)$, $data\_rate(j, q)$, $drop\_rate_{out}(j, q)$) and ($rec\_speed(k, r)$, $drop\_rate_{in}(k, r)$, $data\_rate(k, r)$, $drop\_rate_{out}(k, r)$) are unknown, $j = 1, 2, 3, \ldots, n_q$, $k = 1, 2, 3 \ldots, n_r$. On the other hand, if we can access the $n_q + n_r$ intermediate nodes' network management information through a network management protocol (e.g., Simple Network Management Protocol (SNMP)), then we can retrieve the quadruples ($rec\_speed()$, $drop\_rate_{in}()$, $data\_rate()$, $drop\_rate_{out}()$) from all immediate nodes. So, we further assume that all immediate nodes' management information bases (MIBs) are available, and can be accessed. However, accessing network management information takes time. It is hard to retrieve the desired information for each path in a real time manner right before choosing a primary path. And, before current accurate information is gathered, we cannot make a right decision and choose the right path. On the other hand, if we access the information before choosing the best path, delivery of Q'/Q will be delayed. To solve this problem, we predict the quadruple values for each node by using the exponential average algorithm [31], $\tau_{n+1} = \alpha\tau_n + (1 - \alpha)T_n$, where τ_{n+1} and τ_n are respectively the $(n + 1)$th and nth predicted values of one of the quadruple elements, and T_n is the nth actual value of the feature retrieved from the corresponding MIB. Here,

$$\overline{R_{j,q}}^{n+1} = \alpha_{L_{j,q}} \cdot \overline{R_{j,q}}^{n} + (1 - \alpha_{R_{j,q}}) \cdot R^n_{j,q} \tag{33}$$

$$\overline{M_{j,q}}^{n+1} = \alpha_{M_{j,q}} \cdot \overline{M_{j,q}}^{n} + (1 - \alpha_{M_{j,q}}) \cdot M^n_{j,q} \tag{34}$$

$$\overline{drop\_rate_{in}}^{n+1}(j, q) = \alpha_{dp_{in}(j,q)} \cdot \overline{drop\_rate_{in}}^{n}(j, q) + (1 - \alpha_{dp_{in}(j,q)}) \cdot drop\_rate^n_{in}(j, q) \tag{35}$$

$$\overline{drop\_rate_{out}}^{n+1}(j, q) = \alpha_{dp_{out}(j,q)} \cdot \overline{drop\_rate_{out}}^{n}(j, q) + (1 - \alpha_{dp_{out}(j,q)}) \cdot drop\_rate^n_{out}(j, q) \tag{36}$$

where $\overline{R_{j,q}}^{n+1}$, $\overline{M_{j,q}}^{n+1}$, $\overline{drop\_rate_{in}}^{n+1}(j, q)$ and $\overline{drop\_rate_{out}}^{n+1}(j, q)$($\overline{R_{j,q}}^{n}$, $\overline{M_{j,q}}^{n}$, $\overline{drop\_rate_{in}}^{n}(j, q)$ and $\overline{drop\_rate_{out}}^{n}(j, q)$) are respectively the $(n + 1)$th (the nth) predicted receiving speed, delivery speed, input drop rate and output drop rate of $S_{j,q}$, $\alpha_{R_{j,q}}$, $\alpha_{M_{j,q}}$, $\alpha_{dp_{in}(j,q)}$ and $\alpha_{dp_{out}(j,q)}$ are respectively weights of $S_{j,q}$'s receiving speed, delivery speed, input drop rate and output drop rate, and $R^n_{j,q}$, $M^n_{j,q}$, $drop\_rate^n_{in}(j, q)$ and $drop\_rate^n_{out}(j, q)$ are respectively the nth actual receiving speed,

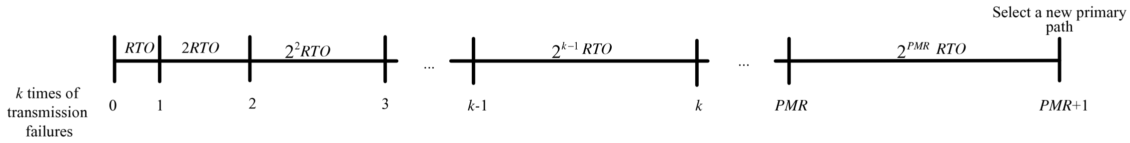

Fig. 4. Timings of k transmission failures.

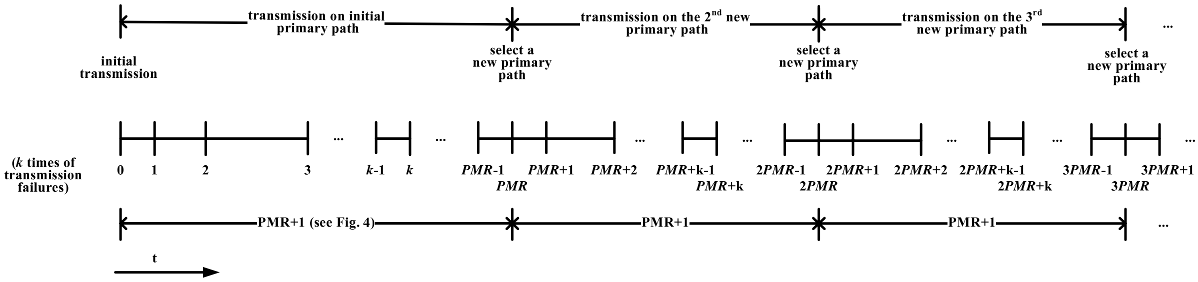

Fig. 5. Timings of transmission/retransmission and primary path selection.

actual delivery speed, actual input drop rate and actual output drop rate. Since SCTP's path change and switch do not occur frequently, we often have enough time to access the actual values of the four terms from each intermediate node's MIB [32]. An example of MIB is the CISCO-IETF-SCTP-EXT-MIB [33].

In an MIB, the items $ipInReceives(t)$ (OID = {1.3.6.1.2.1.4.3}), $ipInDiscards(t)$ (OID = {1.3.6.1.2.1.4.8}), $ipOutRequests(t)$ (OID = {1.3.6.1.2.1.4.10}) and $ipOutDiscards(t)$ (OID = {1.3.6.1.2.1.4.11}) are respectively defined as the accumulated numbers of packets that the underlying router has so far received, dropped on the input side, sent and dropped on the output side since the router started up. By retrieving the four items from intermediate node $S_{j,q}$, $\overline{R_{j,q}}^{n+1}$, $\overline{M_{j,q}}^{n+1}$, $\overline{drop\_rate_{in}}^{n+1}(j,q)$ and $\overline{drop\_rate_{out}}^{n+1}(j,q)$, at time point t_{n+1} can be derived where the $R^n_{j,q}$, $M^n_{j,q}$, $drop\_rate^n_{in}(j,q)$ and $drop\_rate^n_{out}(j,q)$ respectively in Eqs. (33)–(36) can be obtained by accessing the MIB twice at t_{n2} and t_{n1} right after the previous (i.e., the nth) switchover at time t_n, and then expressed by the following equations,

$$R^n_{j,q} = \frac{(ipInReceives(t_{n2}) - ipInDiscards(t_{n2})) - (ipInReceives(t_{n1}) - ipInDiscards(t_{n1}))}{t_{n2} - t_{n1}},$$

$$M^n_{j,q} = \frac{(ipOutRequests(t_{n2}) - ipOutDiscards(t_{n2})) - (ipOutRequests(t_{n1}) - ipOutDiscards(t_{n1}))}{t_{n2} - t_{n1}},$$

$$drop\_rate^n_{in}(j,q) = \frac{ipInDiscards(t_{n2}) - ipInDiscards(t_{n1})}{t_{n2} - t_{n1}},$$

$$drop\_rate^n_{out}(j,q) = \frac{ipOutDiscards(t_{n2}) - ipOutDiscards(t_{n1})}{t_{n2} - t_{n1}}$$

in which $t_{n+1} \gg t_{n2} > t_{n1} > t_n$, $n = 0, 1, 2 \ldots$. Here, $n = 0$ represents the time point right after the SCTP started up (i.e., the time point when the initial primary path has just been selected). Now based on the calculation of CD'_{qr} shown in Eq. (32), we can select a better one from paths q and r. After C_2^{H-1} times of calculation and selection, the new primary path can be found.

3.4. Total cost with retransmission

In the following, besides the total cost of packet delivery without retransmission, we would also like to derive the costs when data due to transmission errors should be retransmitted. To clearly describe the total costs on different numbers of retransmission, we need to consider the two parameters RTO and PMR. In the SCTP, as stated above, when a packet is sent out and the sender cannot receive the corresponding ACK within RTO seconds, the packet will be retransmitted, and as shown in Fig. 4 the $RTO \leftarrow 2 \times RTO$ (i.e., the SCTP duplicates its RTO value). Each time when a packet cannot be successfully delivered within retransmission PMR times (i.e. $PMR + 1$ transmissions), the SCTP will evaluate remaining alternate paths and choose a new primary path [34]. The relationship among RTO, PMR, and the opportunity to choose a new primary is shown in Fig. 5. In fact, Fig. 4 is a part of Fig. 5. When k transmissions (rather than retransmission) fail, the accumulated costs due to timeout are $\sum_{i=1}^{k} 2^{i-1} \cdot RTO$ (=$(2^0 + 2^1 + \cdots + 2^{k-1}) \cdot RTO$). Now, we assume the source node S_0 currently experiences $k - 1$ retransmission failures (i.e., k transmission failures, including the initial transmission failure) and the kth retransmission (i.e., $(k+1)$th transmission) succeeds.

(1) *When* $0 \leq k < PMR + 1$

$k < PMR + 1$ implies the $(k + 1)$th transmission also goes through the initial primary path where the "1" represents the initial transmission. The total cost for successfully delivering Q/Q' at $(k + 1)$th transmission, denoted by T_0, is

$$\begin{aligned} T_0 &= T_{evaluation(0)} + \left(T_{Q\_S}(0) + \sum_{i=1}^{k} 2^{i-1} \cdot RTO + (T_{Q\_processing} - T_{Q\_S}(0) + T_{Q\_T} \right. \\ &\quad \left. + T_{Q\_pro} + T_{Q\_que})\right) + (T_{A\_processing} + T_{A\_T} + T_{A\_pro} + T_{A\_que}) \\ &= T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} \cdot RTO + T_{processing} + T_{Q\_T} + T_{A\_T} + 2(T_{Q\_pro} + T_{Q\_que}) \end{aligned} \tag{37}$$

where $T_{evaluation(i)}$ represents the cost of evaluating all remaining H-i paths of the underlying association. When $i = 0$, $T_{evaluation(0)} = f(H)$. That means the SCTP has to evaluate H paths. The first $T_{Q\_S}(0)$ is the time that the S_0 requires to initially prepare Q'. After Q' is sent out, the *RTO* timer is then initiated. Usually, S_0 keeps Q' in its message buffer until it receives the corresponding *ACK*. So, when Q' for some reason has to be retransmitted, S_0 just retrieves Q' from the buffer without regenerating it again. That is why in Eq. (37) the term $T_{Q\_S}(0)$ is subtracted.

Let $T_{delivery}$ be the time required by the $(k+1)$th transmission which successfully delivers Q' without any retransmission, i.e., $T_{delivery} = TC$ (see Eq. (30))

$$T_{delivery} = T_{processing} + T_{Q\_T} + T_{A\_T} + 2(T_{Q\_pro} + T_{Q\_que}). \tag{38}$$

Then, T_0 in Eq. (37) can be expressed by

$$T_0 = T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} RTO + T_{delivery}. \tag{39}$$

Assume the packet loss rate of the underlying primary path (i.e., the initial primary path) is P_0, which is also Q' retransmission probability. If the occurrence of the ith data delivery failure is denoted by $DF(i)$, based on Bayes' Theorem [35], $P(A|B) = \frac{P(A\cap B)}{P(B)}$, the average delivery cost T_{av} is

$$\begin{aligned} T_{av} &= T_{evaluation(0)} + RTO \cdot P_0 + 2 \cdot RTO \frac{P(DF(2) \cap DF(1))}{P(DF(1))} + 2^2 \cdot RTO \frac{P\left(DF(3) \cap \left(\bigcap_{i=1}^{2} DF(i)\right)\right)}{P(DF(2) \cap DF(1))} \\ &\quad + \cdots + 2^{m-1} \cdot RTO \frac{P\left(DF(m) \cap \left(\bigcap_{i=1}^{m-1} DF(i)\right)\right)}{P\left(\bigcap_{i=1}^{m-1} DF(i)\right)} + \cdots + 2^{k-1} \cdot RTO \frac{P\left(DF(k) \cap \left(\bigcap_{i=1}^{k-1} DF(i)\right)\right)}{P\left(\bigcap_{i=1}^{k-1} DF(i)\right)} \\ &\quad + T_{delivery}(1 - P_0) \\ &= T_{evaluation(0)} + RTO \cdot P_0 + 2 \cdot RTO \frac{P_0^2}{P_0} + 2^2 \cdot RTO \frac{P_0^3}{P_0^2} + \cdots + 2^{m-1} \cdot RTO \frac{P_0^m}{P_0^{m-1}} \\ &\quad + \cdots + 2^{k-1} \cdot RTO \frac{P_0^k}{P_0^{k-1}} + T_{delivery}(1 - P_0) \\ &= T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} \cdot RTO \cdot P_0 + T_{delivery}(1 - P_0) \end{aligned} \tag{40}$$

where $P(DF(i))$ is probability that the ith data delivery failure really occurs, $RTO \cdot P_0$ represents the time on the 1st timeout, and $2^{m-1} \cdot RTO \frac{P\left(DF(m)\cap\left(\bigcap_{i=1}^{m-1} DF(i)\right)\right)}{P\left(\bigcap_{i=1}^{m-1} DF(i)\right)}$ is the time consumed in the mth timeout, $1 \leq m \leq k$.

(2) *When* $k \geq PMR + 1$

$k \geq PMR + 1$ implies that the SCTP has selected the best alternate path as the new primary path r times, $r = 1, 2, \ldots, \left\lfloor \frac{k+1}{PMR+1} \right\rfloor, \ldots$, or H-1, where H is the total number of paths that the underlying association has, $PMR+1$ represents that every $PMR + 1$ failures, a new primary path will be selected, and $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$ means the underlying primary path is the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$th newly selected primary path (excluding the initial path selection). The maximum value of r is H-1 instead of H

also due to excluding the initial. The total cost from when the SCTP starts transmission after an association is established to the time when the $(k+1)$th transmission succeeds is

$$\begin{aligned}
T &= \left(T_{evaluation(0)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO\right) + \left(T_{evaluation(1)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO\right) + \cdots \\
&\quad + \left(T_{evaluation\left(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor - 1\right)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO\right) + \left(T_{evaluation\left(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor\right)} + \sum_{i=1}^{S} 2^{i-1} \cdot RTO + T_{delivery\left(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor\right)}\right) \\
&= \sum_{i=0}^{\left\lfloor \frac{k+1}{PMR+1} \right\rfloor} T_{evaluation(i)} + \left\lfloor \frac{k+1}{PMR+1} \right\rfloor \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO + \sum_{i=1}^{S} 2^{i-1} \cdot RTO + T_{delivery\left(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor\right)} \qquad (41)
\end{aligned}$$

where $S \left(=k - \left\lfloor \frac{k+1}{PMR+1} \right\rfloor (PMR+1)\right)$ is the number of timeouts (i.e., transmission failures) on the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$th primary path before Q' is successfully delivered ($\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$th $=$ 0th means the initial primary path), and there are a total of $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor + 1$ path evaluations (including the initial evaluation). Assume that if there are m remaining paths, then $m + i = H$, and $T_{evaluation}(i) = f(m) = f(H - i)$, which means the cost of the ith evaluation of paths, is proportional to the number of remaining paths where $i = \left\lfloor \frac{k+1}{PMR+1} \right\rfloor$, $0 \le i \le H$ and $T_{delivery\left(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor\right)}$ is the cost required to successfully deliver Q' and receive the corresponding ACK (see Eq. (38)) through the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$th selected primary path. Let $T_{delivery}$ in Eq. (39) be $T_{delivery(0)}$, and $\sum_{i=1}^{0} T_{evaluation(i)} = 0$, then we can conclude that Eq. (41) is the general equation of T. Assume the path failure rate of the ith primary path is P_i which is also retransmission probability of Q' on path i. Let $k_i = k - \left\lfloor \frac{k+1}{PMR+1} \right\rfloor (PMR+1)$ which is the number of timeouts on the ith primary path before Q' is successfully delivered on this path, $i = 0, 1, 2, \ldots, H-1$.

Let c_i be the cost that the SCTP consumes to successfully deliver Q' on the ith primary path. Let $T_{retrans\_time\_out}$ be $\sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO$ which is the total cost of $PMR + 1$ transmission failures, i.e., the SCTP will choose a new path. From Eq. (40), we can derive

$$c_0 = T_{evaluation(0)} + \sum_{i=1}^{k_0} 2^{i-1} \cdot RTO \cdot P_0 + T_{delivery(0)}(1 - P_0)$$

$$c_1 = (T_{evaluation(0)} + T_{retrans\_time\_out}) + T_{evaluation(1)} + \sum_{i=1}^{k_1} 2^{i-1} \cdot RTO \cdot P_1 + T_{delivery(1)}(1 - P_1)$$

$$c_2 = \sum_{i=0}^{1} (T_{evaluation(i)} + T_{retrans\_time\_out}) + T_{evaluation(2)} + \sum_{i=1}^{k_2} 2^{i-1} \cdot RTO \cdot P_2 + T_{delivery(2)}(1 - P_2)$$

$\cdots$

$$c_{H-1} = \sum_{i=0}^{H-2} (T_{evaluation(i)} + T_{retrans\_time\_out}) + T_{evaluation(H-1)} + \sum_{i=1}^{k_{H-1}} 2^{i-1} \cdot RTO \cdot P_{H-1} + T_{delivery(H-1)}(1 - P_{H-1}).$$

The average delivery cost T_{av} can be derived where

$$\begin{aligned}
T_{av} &= \frac{1}{H} \sum_{i=0}^{H-1} c_i \\
&= \frac{1}{H} \left(\sum_{j=0}^{H-2} \sum_{i=0}^{j} (T_{evaluation(i)} + T_{retrans_time_out}) + \sum_{i=0}^{H-1} T_{evaluation(i)} \right. \\
&\quad \left. + \sum_{h=0}^{H-1} \sum_{i=1}^{k_h} 2^{i-1} \cdot RTO \cdot P_h + \sum_{i=0}^{H-1} T_{delivery(i)}(1 - P_i) \right). \qquad (42)
\end{aligned}$$

4. Experimental results

4.1. Simulation environment setup

Our simulations were carried out by running a revision of Delaware University's SCTP module [36] for NS-2 [37]. The simulation topology as shown in Fig. 6 includes two end nodes, sender and receiver, both of which have 4 IP addresses. Routers 1–1 and 1–2, routers 2–1 and 2–2, …, and routers 4–1 and 4–2 are set up between the two end nodes. Router i–1 is connected to router i–2 and $i = 1, 2, 3, 4$. The SCTP parameters are all default values. Other parameters are listed in Table 4. The sender continuously sends 2 Mbps FTP data to the receiver. Switchover occurs at the 10th second. Five experiments

Table 4
Simulation parameters.

Parameters	Value
Sending rate	2 Mbps FTP data
Propagation delay	50 ms
SCTP chunk size	1468 bytes
SCTP MTU	1500 bytes
Path 1's bandwidth	2 Mbps
Path 2's bandwidth	1.5 Mbps
Path 3's bandwidth	1.8 Mbps
Path 4's bandwidth	1 Mbps

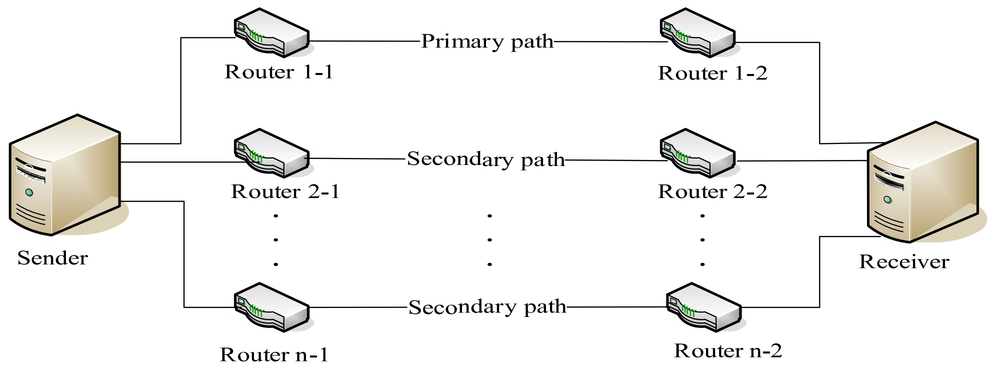

Fig. 6. Simulation topology ($n = 4$).

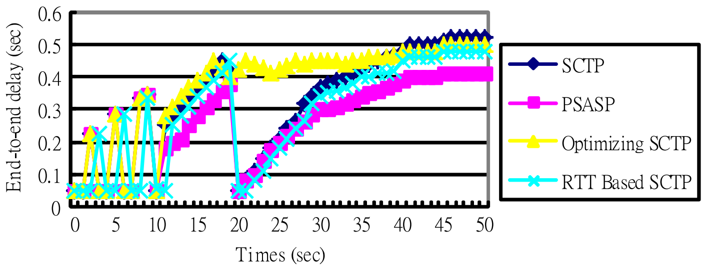

Fig. 7. End-to-end delays of the four tested schemes.

were performed in this study. The first evaluated the PSASP's four QoS parameters, including end-to-end delays, jitters, throughputs and packet drop rates. The second redid the first experiment given different numbers of routers along a tested routing path. The third also redid the first experiment but each tested path was given different error rates. The fourth measured the scales of delays. The fifth evaluated switching costs when H paths are given. All end-to-end delays in the following are calculated by involving Eq. (30).

4.2. Simulation results of the first experiment

In the first experiment, three state-of-the-art systems, including the standard SCTP [15], Optimized SCTP [8], and RTT based SCTP [4], were tested and compared with the PSASP. The default primary path of the standard SCTP was set to path 2.

The experimental results of the four schemes for the end-to-end delays as illustrated in Fig. 7 were initially almost the same. But, after the first switchover, the PSASP had less delays than others, and right after the switchover, the end-to-end delays of the four systems between the 11th and 12th seconds due to shrunken congestion window did not increase sharply because they all had enough bandwidth to transmit packets. When time passed and more packets and overheads were sent and involved, respectively, the delays increased quickly. But the PSASP had less delay because it selected the best path. The Optimized SCTP as stated above reduced its congestion window size slowly. Due to serious congestion, its end-to-end delays were then longer. On the other hand, the *cwnd* size of RTT based SCTP is not reduced on packet drop [4]. Its adjustment only depends on the measured RTT. So, the delay was lower than the standard SCTP. The PSASP calculates path delays to select the fastest path, but the standard SCTP, and Optimized SCTP do not specify how to select alternate paths as the primary paths. Their selection is based on the order in which the paths were specified when the underlying association was established.

In the best case, when the fastest path is the first alternate path, the second fastest is the second alternate path ..., and the slowest one is the last path, then the order of the path selection of the four tested schemes will be the same.

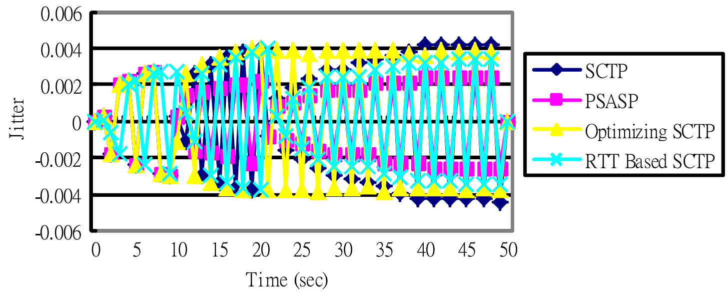

Fig. 8. Jitters of the four tested schemes.

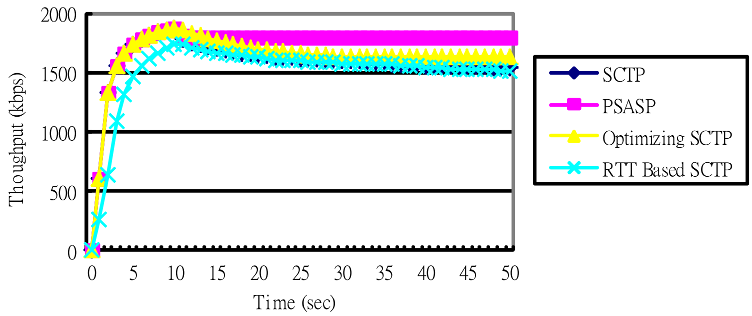

Fig. 9. Throughputs of the four tested schemes.

Table 5
Packet loss rates of the four tested schemes.

Protocol	No. of packets sent	No. of packets received	No. of packets lost	Packet loss rate (%)
SCTP	22 826	22 724	102	0.446
PSASP	27 077	27 010	67	0.247
OPT SCTP	23 543	23 390	153	0.649
RTT SCTP	22 483	22 402	81	0.360

However, this cannot discriminate the characteristics of path selection. The Optimized SCTP and RTT based SCTP adjust their sizes of congestion windows with specific methods when necessary. So when congestion is not severe and the size of the congestion window does not need to be hugely reduced, the two schemes are better than the PSASP and standard SCTP. But this situation is not always true. We should consider the general case in which congestion is or is not severe and the current default path is or is not the fastest one. Hence, a method that can keep the association performing the best is required. Choosing the best path and adjusting the size of the congestion window are the solutions. The PSASP follows the standard SCTP's method to adjust the size of the congestion window. However, the effect of adjusting window size due to limited bandwidth of the current path is sometimes not significant. In fact, if the bandwidth of the current path is wide, the probability of switching over to an alternate path due to low packet drop rate will also be lower. That is why the PSASP's delays as shown in Fig. 7 are relatively shorter.

Fig. 8 shows that the PSASP had smaller jitters than the others had. At the point when the primary path began its transmission, the jitters vibrated because the two sides of the path need to exchange information, e.g., four-way handshake, resulting in more transmission overheads. However, the transmission and jitters were soon stable. When switchover occurred, the jitters vibrated again, and the other three schemes' jitters are larger than they were. Generally, longer transmission delays result in larger jitters [38], and lower traffic often causes shorter and smoother vibration. The PSASP had a similar phenomenon, but the vibration was smoother and smaller since the PSASP always chooses the path currently with the widest bandwidth as the new primary path. The Optimized SCTP's vibration was huge because its congestion window decreases slowly even when traffic is seriously congested.

Fig. 9 illustrates experimental results for throughputs. Before switchover, throughputs of the four schemes were not significantly different. After the switchover, since the path with the shortest RTT was selected, the PSASP outperformed the others. Although Optimized SCTP and RTT based SCTP adjust their congestion window, their performance is limited also by the current path's bandwidth.

Table 5 lists experimental results of packet loss rates. The PSASP exhibits the best due to choosing the best alternate path (i.e., path 3) which provides a higher transmission quality and stabler environment than the default path (i.e., path 2) does. A wider-bandwidth path can transmit and process many more packets and reduce the probability of network congestion so as to decrease the packet loss rate and probability of packet retransmission.

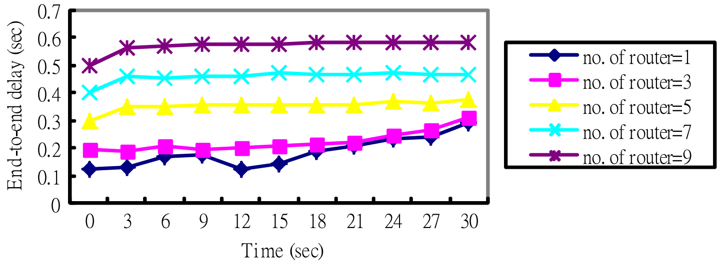

Fig. 10. End-to-end delays of the PSASP given different numbers of routers.

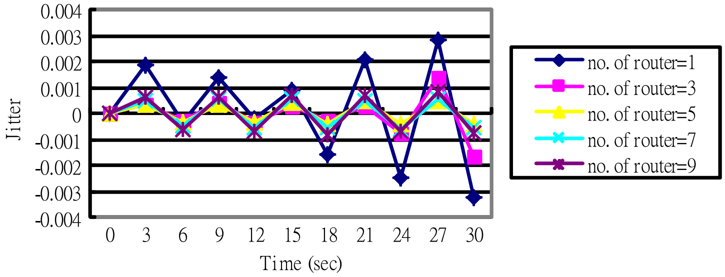

Fig. 11. Jitters of the PSASP given different numbers of routers.

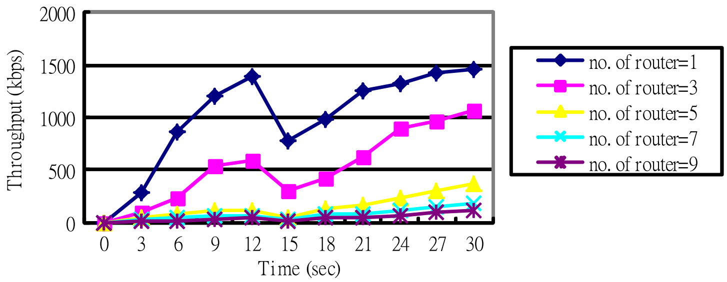

Fig. 12. Throughputs of the PSASP given different numbers of routers.

Generally, the Optimized SCTP has better throughputs than standard SCTP has, since when packets are lost, the size of its congestion window shrinks slowly. But this often causes a high packet loss rate. Its delays and jitters due to packet retransmission are also relatively huge. The RTT based SCTP sacrifices a portion of its throughputs by frequently adjusting the size of the congestion window to exploit lower delays, jitters and packet loss rates than those of the standard SCTP.

4.3. Performance on different numbers of routers

Fig. 10 illustrates the experimental results for end-to-end delays of the second experiment. The numbers of routers given are 1, 3, 5, 7 and 9. When the number of routers increases, the end-to-end delays are obviously longer, since accumulated transmission, propagation and queuing delays are all longer. Congestion occurs at the 15th second, making longer delays for number of routers = 1 and 3. But, the influence on the other three numbers of routers is not significant because when a node, e.g., node *i*, is congested, the sender's congestion window is not reduced immediately. Node *i*'s downstream nodes continue transmitting packets originally queued in their message buffers to their immediate downstream nodes. Node *i*'s upstream nodes keep queuing packets in their message buffers. Before the downstream nodes' buffers are all empty and upstream nodes' buffers are all full, congestion on node i may no longer exist. So, it can again supply enough packets for downstream nodes to continue their transmission, and to relay enough packets for its upstream nodes. We call this phenomenon packet-flow regulation. A path with many more routers has a better regulation effect since many more packets can be accumulated in the message buffers.

Fig. 11 illustrates experimental results for jitters. Jitters are relatively huge when number of routers = 1, especially after the 15th second because of network congestion. But, when number of routers = 5, 7 and 9, jitters are not significantly affected, also due to packet-flow regulation. Generally, as stated above, longer transmission delays result in larger jitters. In this case, the number of routers = 1 which has a smaller packet-flow regulation effect yields higher jitters.

The experimental results for throughputs are illustrated in Fig. 12. We can see that the larger the number of routers, the lower the performance because packets flow through more routers producing many more unnecessary overheads. That is,

Table 6
Packet loss rates given different numbers of routers.

No. of routers	No. of packets sent	No. of packets received	No. of packets lost	Packet loss rate (%)
1	26 375	26 305	70	0.265
3	24 276	24 206	70	0.288
5	22 140	22 073	67	0.302
7	16 123	16 070	53	0.328
9	12 597	12 548	49	0.388

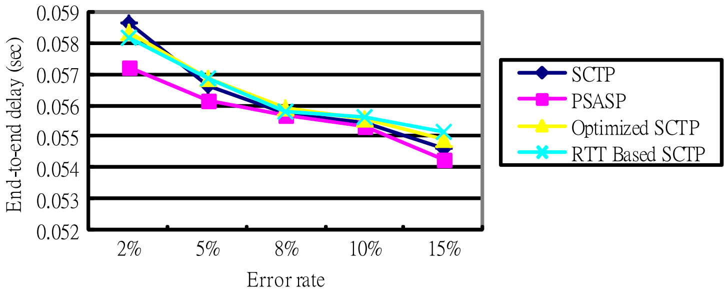

Fig. 13. End-to-end delays of the four tested schemes given different error rates.

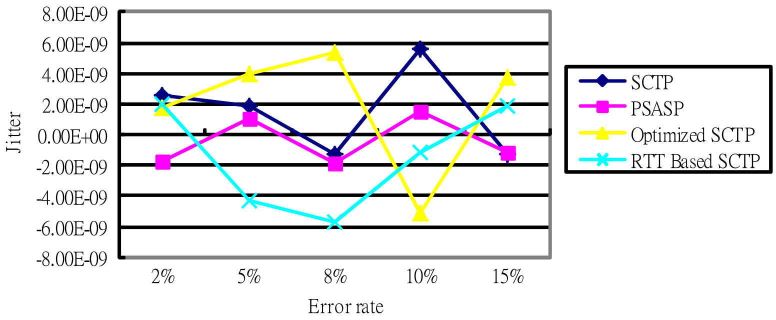

Fig. 14. Jitters of the four tested schemes given different error rates.

when a packet P arrives at a router R, P will enter R's message queue/buffer and wait to be processed and transmitted. More routers result in longer accumulated queuing and transmission delays. Furthermore, many more routers also cause a higher packet loss rate. Hence, the performance is always lower.

The experimental results for packet loss rates are listed in Table 6. When the number of routers increases, packets are transmitted though more nodes, resulting in higher drop rates, and of course higher packet loss rates. Basically, if there are n routers on a path and their drop rates are respectively $P_1, P_2, P_3, \ldots,$ and P_n, then the probabilities that a packet can be successfully delivered by them is denoted by P_S, $P_S = \prod_{i=1}^{n}(1 - P_i)$. A larger n will yield a smaller P_S.

Now, we can conclude that many more routers, e.g., n routers, will cause more transmission overheads since a packet when passing through a router has to wait to be processed and sent, and the packet has to propagate and be transmitted $n + 1$ times. An ACK has similar phenomena. Both increase the total waiting time and degrade the performance.

4.4. Performance on different error rates

In the third experiment, we evaluate the four tested systems given different error rates, including 2%, 5%, 8%, 10% and 15%, to see how error rates affect a system. We can see the end-to-end delays of the four schemes as shown in Fig. 13 are not significantly different. Due to selecting the path with the shortest delay, the end-to-end delays of the PSASP are shorter than those of others since the path with wider bandwidth and better transmission performance can decrease the queuing delay [4] and end-to-end delays.

Fig. 14 illustrates the experimental results for average jitters. We see that the four schemes have different ranges of jitters. But, due to choosing the path with less delay the PSASP's jitters are the smallest compared to other schemes', thus suitable for transmitting multi-media and audio data.

Fig. 15 illustrates the experimental results for average throughputs. When the error rates increase, the performances decrease sharply. But, the PSASP outperforms the others. In theory, when the error rates increase from, e.g., 2%, to, e.g., 15%, the throughputs will decrease from 98% to 85%. But, the resulting throughputs are rather small. Since each time when a

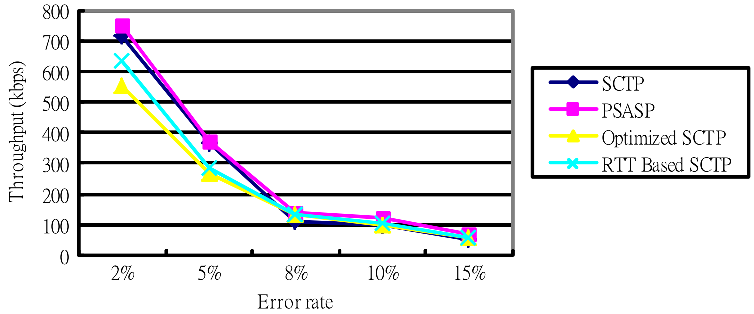

Fig. 15. Throughputs of the four tested schemes given different error rates.

Table 7
The scale of four delays.

Delays	Scale (s)
Average processing delay	0.00629
Average transmission delay	0.005872
Average propagation delay	0.05
Average queuing delay	0.20353
Average end-to-end delay	0.266836

Table 8
Average queuing delay with different numbers of routers.

No. of routers	Avg. queuing delay (s)	Avg. end-to-end delay (s)	Percentage (%) (= Avg. queuing delay/Avg. end-to-end delay)
1	0.21206	0.26906	78
3	0.23038	0.28538	80
5	0.302201	0.36820	82
7	0.414312	0.476312	86
9	0.526548	0.586548	89

packet is retransmitted due to loss, the RTO increases doubly. Hence, a packet transmitted on a high error-rate path needs to wait for a longer RTO time, thus lowering throughputs.

Lastly, we can conclude that higher error rates cause higher packet loss rates, many more retransmitted packets and many more overheads. Basically, when many ACKs are lost, the corresponding data packets will be retransmitted at least twice, consequently consuming wider bandwidth, causing longer end-to-end delays, and resulting in worse throughputs and jitters.

4.5. The scale of four delays

Table 7 shows that the average processing delay and transmission delay of the PSASP are about 5 and 6 ms, respectively. Their scales are relatively smaller than those of the other two because their performance heavily depends on hardware speed and bandwidth of network interfaces. Processing data with hardware can often obtain very good performance. The average propagation delay and queuing delay are about 50 ms and 200 ms, respectively. The latter is close to our measured average end-to-end delay. Now, it is clear that the bottleneck of data transmission is queuing delay. This meets what [8] mentioned. When the queue is full, the following packets will be dropped. The longer waiting time will cause a longer end-to-end delay. This phenomenon is also true for the second experiment, i.e., the larger numbers of routers result in a higher queuing delay. Table 8 also lists the measured queuing delays given different numbers of routers. We can see that when the number of routers increases, the queuing delays are also higher. Our conclusion is that percentages of queuing delays roughly range between 75% and 90% of end-to-end delays.

4.6. Switching cost of H paths

In this experiment, we evaluated the path's switch cost. Switchover occurs at the 10th second, and we measured the time period from when the tested system started to the time point when the secondary path's first packet was successfully delivered and processed. The average time measured was 10.01008 s, indicating that switchover costs about 10 ms. In our simulation environment, when the primary path fails, the PSASP calculates the remaining H-1 paths' cost differences. If we compare paths by pair, the time of comparison of the remaining H-1 paths to select the fastest path is H-2. The average cost for evaluating the cost difference of two paths is about 5 ms. If $H = 10$, the cost will be 50 ms. So, when H is higher, the evaluation cost will also be increased. The SNMP packet format [39] consists of three parts, including SNMP Header (11 bytes), SNMP PDU Header (12 byte) and PDU data. The PDU data comprises OID length and OID data. We only need to access the four OIDs (i.e., *rec_speed()*, $drop\_rate_{in}()$, *data_rate()*, $drop\_rate_{out}()$) from the concerned MIB. A PDU data is 21 bytes for each OID access, i.e., an SNMP packet is 44 bytes in length. So, the number of bytes delivered for the four OIDs is 176 ($=44 * 4$) bytes, which is very small compared to the data packet size (1468 bytes). For example, if the bandwidth of a path is 2 Mbps, it can transmit 250 000 bytes per second. The percentage of SNMP packets generated on each MIB retrieval from the n routers is $176 * n/250\,000$. Let us consider the Time to Live (TTL) value given by a default router of a host [40]. The Linux and FreeBSD systems adopt 64 as the initial TTL value, which means the SNMP packets at most increase 0.044544% of network traffic. Of course, this is negligible. Now, we can conclude that the SNMP packets do not significantly increase occupied bandwidth and the following data transmission.

5. Conclusions and future research

In this study, we develop a new path selection and switching scheme for the SCTP, the PSASP, which considers the key path performance influential factor, i.e., round-trip delay, to select a primary path for the SCTP so as to provide the SCTP network transmission with wider bandwidth and a more reliable environment. The round-trip delay is the time required to successfully deliver a packet and receive the corresponding ACK. We further decompose the round-trip delay into processing, transmission, propagation and queuing delays, and analyze the influential factors of the four delays.

We also consider the PSASP's retransmission costs on different retransmission counts (i.e., $k < PMR+1$ and $k \geq PMR+1$) and different paths' packet loss rates where a packet's loss rate is also the path's transmission failure probability. This helps us to infer the average costs of packet delivery and retransmission. Experimental results show that the PSASP can accurately evaluate performance of alternate paths so as to select the one with the widest bandwidth as the primary path. This is why the PSASP outperforms the other three tested schemes.

In the future, we would like to derive the PSASP's mathematical model of reliability which is a formal model, so that a user can determine the reliability of the PSASP before using it. We will also study how the parameters considered affect the arrival and service rates of a path segment, so that we can more preciously estimate the performance of a path. The purpose is to accurately estimate queuing delay. These questions constitute our future research.

References

[1] E.P. Ribeiro, V.C.M. Leung, Minimum delay path selection in multi-homed systems with path asymmetry, IEEE Communications Letters 10 (3) (2006) 135–137.
[2] F.Y. Leu, A novel network mobility handoff scheme using SIP and SCTP for multimedia applications, Journal of Network and Computer Applications 32 (2009) 1073–1091.
[3] J. Noonan, P. Perry, S. Murphy, J. Murphy, Simulations of multimedia traffic over SCTP modified for delay-centric handover, in: World Wireless Congress, 2004.
[4] M. Dahal, D.Kr. Saikia, RTT based congestion control and path switching scheme for SCTP, in: The International Conference on Communication Technology, 2006, pp. 1–4.
[5] A. Kelly, G.M. Muntean, P. Perry, J. Murphy, Delay-centric handover in SCTP over WLAN, IEEE Transactions on Automatic Control and Computer Science 49(63) (4) (2004) 1–6.
[6] C.Z. Yang, W.K. Chang, I.H. Huang, CS-SCTP: a collaborative approach for secure SCTP over wireless networks, in: The IEEE International Region 10 Conference, 2007, pp. 1–4.
[7] P. Behbahani, V. Rakocevic, J. Habermann, nSCTP: a new transport layer tunnelling approach to provide seamless handover for moving networks, in: The International Conference on Mobile Wireless Communications Networks, 2007, pp. 71–75.
[8] A. Al-kaisan, Md. Ashrafuzzaman, S.M.M. Ahsan, Reducing congestion collapse and promoting fairness in the internet by optimizing SCTP, in: IEEE Computer and Information Technology, 2008, pp. 1–5.
[9] Information technology—Open system interconnection basic reference model: the basic model, ISO/IEC Std. 7498-1, 1994.
[10] F.Y. Leu, Y.L. Chen, A path switching scheme for SCTP based on round trip delay and under the assistance of network management schemes, in: The International Conference on Computer Science and its Applications, 2009, pp. 103–108.
[11] H.C. Kim, Y.H. Kim, K.J. Kim, J.W. Chung, High-performance data transfer using SCTP-based compact association scheme, in: The International Conference on Computational Science and its Applications, 2007, pp. 389–398.
[12] M.M. Monowar, M.O. Rahman, C.S. Hong, Multipath congestion control for heterogeneous traffic in wireless sensor network, in: The International Conference on Advanced Communication Technology, 2009, pp. 1711–1715.
[13] S. Fallon, P. Jacob, Y.S. Qiao, L. Murphy, E. Fallon, A. Hanley, SCTP switchover performance issues in WLAN environments, in: The IEEE International Conference on Consumer Communications and Networking Conference, 2008, pp. 564–568.
[14] H. Hassan, M. El-Shehaly, A. Abdel-Hamid, Routing and reliable transport layer protocols interactions in MANETs, in: The International Conference on Computer Engineering and Systems, 2007, pp. 359–364.
[15] Internet Engineering Task Force (IETF), RFC 2960 (Stream Control Transmission Protocol), 2000. http://www.faqs.org/rfcs/rfc2960.html.
[16] P. Natarajan, R.I. Janardhan, D.A. Paul, R. Stewart, SCTP: an innovative transport layer protocol for the web, in: The International World Wide Web Conference, ACM Press, New York, NY, 2006.
[17] S.J. Koh, M.J. Chang, M. Lee, mSCTP for soft handover in transport layer, IEEE Communications Letters 8 (3) (2004) 189–191.
[18] S. Fu, M. Atiquzzaman, W. Ivancic, SCTP over satellite networks, in: IEEE Workshop on Computer Communications, 2003, pp. 112–116.
[19] S.T. Kim, S.J. Koh, Y.J. Kim, Performance of SCTP for IPTV applications, in: The Advanced Communication Technology International Conference, vol. 3, 2007, pp. 2176–2180.
[20] A.Z.M. Shahriar, M. Atiquzzaman, W.D. Ivancic, Evaluation of the route optimization for NEMO in satellite networks, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2 (2) (2011) 46–66.
[21] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers, in: ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234–244.
[22] C. Perkins, E. Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, in: IETF RFC 3561, July 2003.
[23] D. Johnson, D. Maltz, Y.C. Hu, The dynamic source routing protocol for mobile ad hoc networks for IPv4, in: IETF RFC 4728, February 2007.

[24] R. Stewart, et al. Stream Control Transmission Protocol, IETF RFC 2960, October 2000.
[25] R. Ohlendorf, A. Herkersdorf, T. Wild, FlexPath NP: a network processor concept with application-driven flexible processing paths, in: The International Conference on Hardware/Software Codesign and System Synthesis, 2005, pp. 249–284.
[26] H.Y. Kim, S. Rixner, Performance characterization of the FreeBSD network stack, Rice University Computer Science, Technical Report TR05-450, June 2005.
[27] National Institute of Standards and Technology, Federal Information Processing Standard Publication 197, The Advanced Encryption Standard (AES), November 2001.
[28] K. Kamphenkel, M. Blank, J. Bauer, G. Carle, Adaptive encryption for the realization of real-time transmission of sensitive medical video streams, in: The International Symposium on World of Wireless, Mobile and Multimedia Networks, 2008, pp. 1–6.
[29] J. Fox, Applied Regression Analysis, Linear Models, and Related Methods, Sage Publications, 1997.
[30] D. Gross, C.M. Harris, Fundamentals of Queuing Theory, third ed., Wiley-Interscience, 2005.
[31] Exponentially weighted moving average (EWMA), June 3, 2009. http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm.
[32] Internet Engineering Task Force (IETF), RFC 4293 (Management Information Base for the Internet Protocol (IP)), 2006. http://tools.ietf.org/html/rfc4293.
[33] CISCO-IETF-SCTP-EXT-MIB, January 27, 2010. http://www.oidview.com/mibs/9/CISCO-IETF-SCTP-EXT-MIB.html.
[34] Y. Qiao, E. Fallon, J. Murphy, L. Murphy, X. Zhu, G. Hayes, A. Matthews, A. Hanley, Performance analysis of multi-homed transport protocols with network failure tolerance, IET Communications 2 (2) (2008) 336–345.
[35] Bayes' Theorem, October 15, 2011. http://plato.stanford.edu/entries/bayes-theorem/.
[36] A. Caro, J. Iyengar, ns-2 SCTP module, Version 3.5. http://www.armandocaro.net/software/ns2sctp/.
[37] UC Berkeley, LBL, USC/ISI, and Xerox Parc, ns-2 documentation and software, Version 2.33, May 31, 2008. http://www.isi.edu/nsnam/ns.
[38] F.Y. Leu, Y.T. Huang, Maximum capacity in chain-topology wireless mesh networks, in: Wireless Telecommunications Symposium, 2008, pp. 250–259.
[39] R.Z. Lin, A. Liao, H.C. Chao, Implementing SIP-based technology for management framework, in: The International Conference on Mobile Technology, Applications and Systems, 2008.
[40] K. Fujii, S. Goto, Correlation between hop count and packet transfer time, in: APAN/IWS2000, 2000.